**NEC: An Open Source Framework for Name Data Networking, Edge Computing, and Cloud Computing**

Rehmat Ullah1,Muhammad Atif Ur Rehman2,Byung-Seo Kim3

12Department of Electronics and Computer Engineering, Hongik University, Republic of Korea

3Department of Computer and Communications Engineering, Hongik University, Republic of Korea

1rehmat\_ciit@hotmail.com, 2atif\_r@outlook.com, 3jsnbs@hongik.ac.kr

**Summary**

Named data networking (NDN) and edge cloud computing (ECC) are emerging technologies that are considered as the most representative technologies for the future Internet. Both technologies are the promising enabler for the future Internet such as fifth generation (5G) and beyond which requires fast information response time. We believe that clear benefits can be achieved from the interplay of NDN and ECC and enables future network technology to be much more flexible, secure and efficient. In this project, therefore, we integrate NDN with ECC in order to achieve fast information response time. Our framework is based on N-Tier architecture and comprises of three main Tiers. The NDN is located at the Tier1 (Things/end devices) and comprises of all the basic functionalities that connect Internet of Things (IoT) devices with Tier 2 (Edge Computing), where we have deployed our Edge node application. The Tier 2 is then further connected with Tier 3 (Cloud Computing), where our Cloud node application is deployed at multiple hops on the Microsoft Azure Cloud machine located in Virginia, WA, USA.

**Proposed Framework**

Our proposed framework is based on N-Tier architecture and comprises of 3 main tiers: Tier 1, Tier 2 and Tier 3 and represents Things/end devices, Edge computing and cloud computing respectively. Tier 2 and Tier 3 is then further divided into layers.

**TIER1: THINGS/END DEVICES**

This Tier is involved with end user(s) or IoT device(s) directly. There may be several different types of IoT devices/users coexisting, such as sensor nodes, end user’s smart hand-held devices (smart phone, smart watch, smart vehicles etc.). These device(s) request for data and services and are connected at single hop to the edge Tier.

**TIER2: EDGE COMPUTING**

The edge computing Tier comprises of edge servers and provide delay constrained service request to end users. This Tier is located at one hope distance from the Things Tier. The edge computing Tier is not like the actual cloud computing Tier, however, provides medium number of service request with low latency.

**TIER3: CLOUD COMPUTING**

The cloud computing Tier is the topmost Tier and comprises of traditional cloud servers and has enough storage and computing resources. This Tier is located at multiple hops from the Things Tier and edge computing Tier. However, this Tier comes with a cost of higher latency to the end users.

**Proposed Architecture Details**

Figure 1 illustrates a detailed architectural diagram of the proposed framework. In the following subsections we have provided the detail of our proposed framework and are discussed as follows:

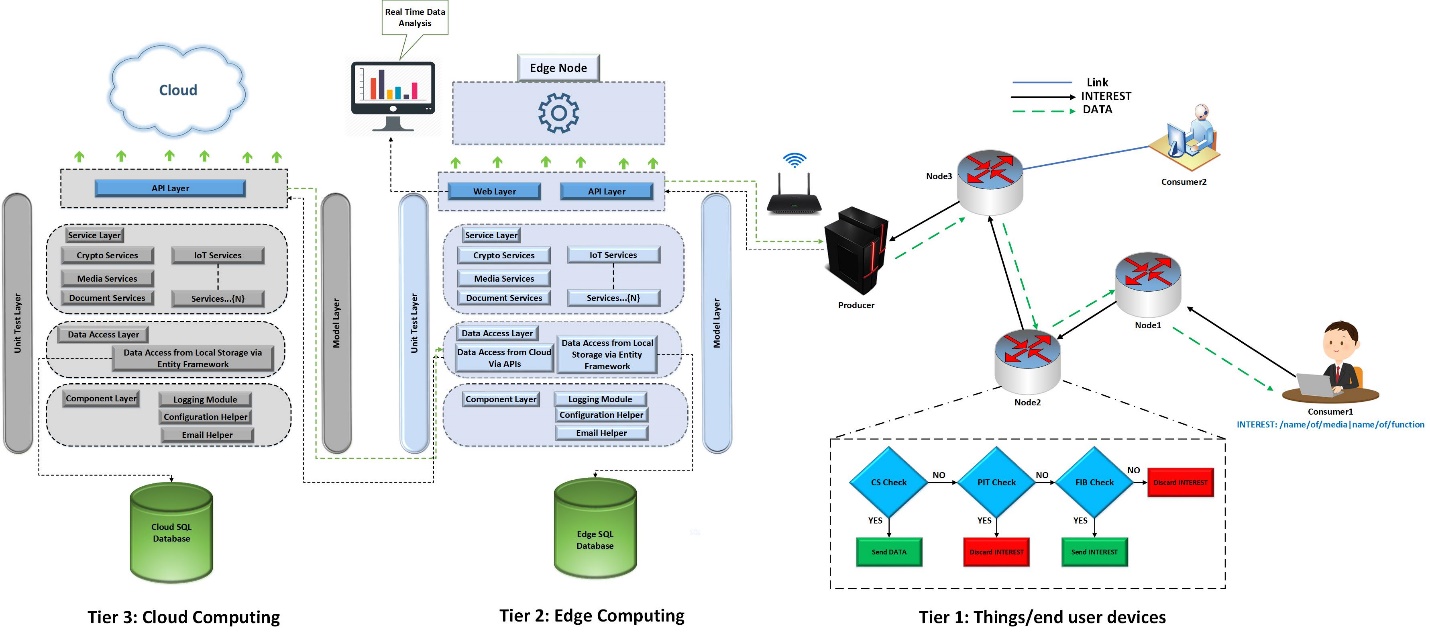


FIGURE 1. An architectural diagram of the proposed framework.

**TIER 1: Named Data Networking**

At this layer we have used the NDN technology where all the communication is content centric, and content are requested at the network layer using interest names. At the Things Tier, without loss of generality, we have used a simple scenario of three nodes i.e. consumer node, relay node and the producer node (gateway). In this scenario, the consumer node send request for content and/or services via relay node. If the relay node has the services/content cached before, then it will send back to the user as the conventional NDN architecture does. Otherwise the request for content/service will be forwarded to the gateway node. The gateway node possibly has the content and also maintains a list of services. In order to better differentiate the gateway node of NDN, edge node and the cloud node, we assume that in our framework the gateway node maintains a limited number of services for end users. The edge node is power full than gateway node of NDN where a large amount of services is listed and similarly the cloud is resource enrich platform for all type of resources.

**TIER 2: Edge Computing**

In our proposed framework, the Tier 2 represents the edge computing where our edge node application is deployed. We further divide the edge computing Tier into seven different layers and each layer is discussed in detail as follows.

The API layer (Layer 1) is responsible to satisfy the requests from NDN network (ndnSIM). The structure of our proposed.NET web API is depicted in Figure 2. The web layer (Layer 2) shows a graphical user interface for real time data display. Service Layer (Layer 3) coordinates data and services between an API layer and the Data Access Layer (DAL) (Layer 4). It contains all the business logics of the services on which the decision might be taken. DAL is responsible for the data management, typically using a database such as SQL, MySQL, Oracle, MongoDB, etc. In our implementation, we have used SQL database. The unit test layer (Layer 5) is provided in order to ease the testing of an application. In component layer (Layer 6) we

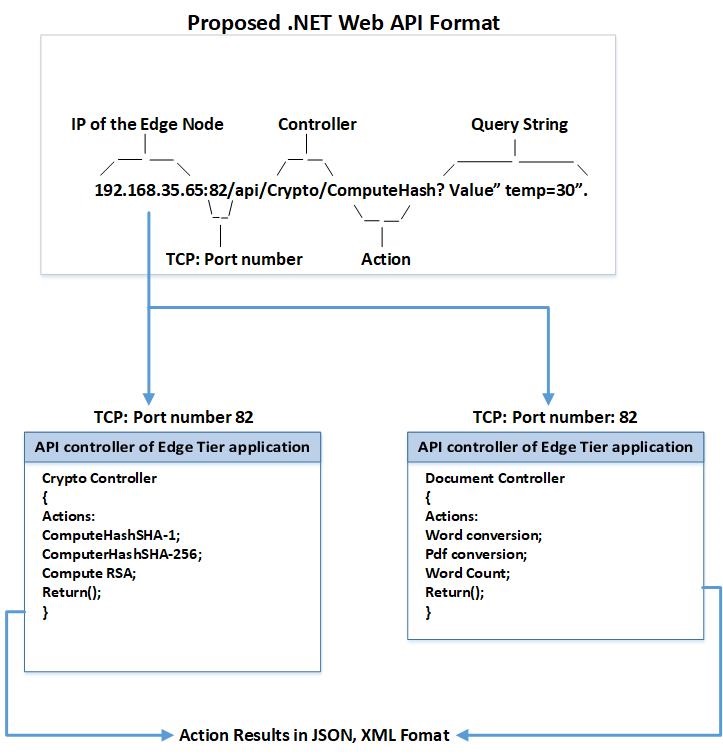


FIGURE 2. Proposed.NET web API for IoT services.

have provided those code modules that are shared among different layers. Model layer (Layer 7) comprises of various type of classes used for carrying data among different layers.

**TIER 3: Cloud Computing**

The cloud node comprises of all the layers which are detailed in the edge Tier except the web layer and the subpart of the data access layer where request is sent to the cloud for data access (since we don’t need to get data farther than the cloud). We deployed our cloud node at multiple hops at the Microsoft Azure Windows server 2016 and for the cloud storage we have used SQL 2017 developer version. Any database can be used in this framework for storage such as Oracle, My SQL, MongoDB etc. In the evaluation section we have provided the complete details about the cloud server specifications.

**Experimental Setup**

The testbed includes Edge node machine, access point, and Microsoft Azure Cloud server machine. The detailed hardware and software configurations are as follows: For NDN network we generate requests from ndnSIM to the edge node. In order to do that we modified some of the code in ndn-producer.cpp file and ndn-producer.hpp file. Our custom function which generates the request is using boost/asio.hpp library and it’s class is asio::ip::tcp. ndnSIM is running on Linux using VMware. The VMware has 8 GB RAM and 4 core CPU. Edge application is hosted at one hop from NDN network on a system with specifications of 16 GB RAM, core-i7, 4710HQ-CPU and @ 2.40 Ghz core. We have developed our edge application using .NET framework. For the deployment of edge application, we have used Internet Information Services (IIS). IIS is a flexible, general-purpose web server provided by Microsoft. The IIS web server accepts requests from remote client computers and returns the appropriate response. This basic functionality allows web servers to share and deliver information across local area networks, such as corporate intranets, and WANs such as the Internet. A web server can deliver information to users in several forms, such as static webpages coded in HTML; through file exchanges as downloads and uploads; and text documents, image files, JSON, XML and more. For cloud application we have used Microsoft Azure Window Server 2016 Data Centre with specification of 8 GB RAM, Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30Ghz 2.29Ghz. Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications

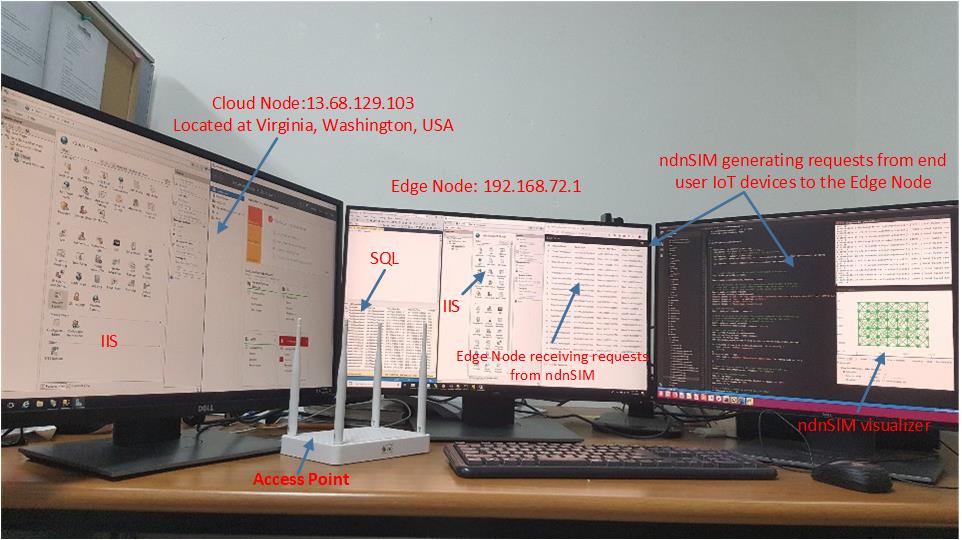


FIGURE 3. NDN based edge cloud computing testbed.

and services through a global network of Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems. Cloud application is also developed in.NET framework and deployed using IIS server. For the database of the edge node and the cloud node we have used SQL 2017 developer version. Our NDN based edge computing testbed is shown in Figure 3. It is to be noted that we have used a limited version of Cloud VM with limited amount of resources for testing our system. However, these resources were enough for our experiments. Since we need the Cloud VM just for testing the latency related measurements and to show the real cloud behavior at multiple hops. Therefore, high amount of resources was not necessary for our evaluations. We testify our system with lower amount of Cloud VM resources and that were enough for our measurements. The system is completely scalable; however, it depends on the user requirements. If one’s wants to scale the system at large scale, then they could scale it easily just by buying the Cloud VM according the requirements. Figure 3 shows our experimental setup and testbed for NDN based edge cloud computing.

**Required Skills**

NDN: ndnSIM, EDGE & Cloud: ASP.NET, Entity Framework, SQL, MongoDB, Razor, JavaScript, and jQuery.

After the **ACM SIGCOMM 2018 hackathon**, we actively worked on our project and as a result we have published a full paper of our framework in *IEEE Access Journal* titled “**Design and Implementation of an Open Source Framework and Prototype For Named Data Networking based Edge Cloud Computing System”.**

The full paper of the framework can be accessed at <https://ieeexplore.ieee.org/abstract/document/8703380>

**Publications**

We have published the following papers from our project since the hackathon and are listed below.

1. R. Ullah, M. A. U. Rehman and B. Kim, "Design and Implementation of an Open Source Framework and Prototype For Named Data Networking-Based Edge Cloud Computing System," in IEEE Access, vol. 7, pp. 57741-57759, 2019. doi: 10.1109/ACCESS.2019.2914067.
2. Rehmat Ullah, Muhammad Atif Ur Rehman, and Byung Seo Kim. 2019. Poster: A Testbed Implementation of NDN-based Edge Computing For Mobile Augmented Reality. In Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19). ACM, New York, NY, USA, 181-181. DOI: <https://doi.org/10.1145/3301293.3309565>
3. Muhammad Atif Ur Rehman, Rehmat Ullah, and Byung-Seo Kim, “Caching in Named Data Networking with an Open Source Edge Computing Framework” Korean Network Operations and Management (KNOM) Conference, May-30~31, 2019, Daegu, Korea.

**Code Repositories**

**NDN:**

1. <https://github.com/atifrehman/NEC/tree/master/ndnSIM/apps>
2. <https://github.com/Rehmatkhan/NEC/tree/master/ndnSIM/apps>

**Edge:**

1. <https://github.com/atifrehman/NEC/tree/master/Edge>
2. <https://github.com/Rehmatkhan/NEC/tree/master/Edge>

**Cloud:**

1. <https://github.com/atifrehman/NEC/tree/master/Cloud>
2. <https://github.com/Rehmatkhan/NEC/tree/master/Cloud>